Differential Evolution with Reversible Linear Transformations

Tomczak, Jakub M.; Wglarz-Tomczak, Ewelina; Eiben, Agoston E.

published in
GECCO '20
2020

DOI (link to publisher)
10.1145/3377929.3389972

document version
Publisher's PDF, also known as Version of record

Differential Evolution with Reversible Linear Transformations

Jakub M. Tomczak
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
j.m.tomczak@vu.nl

Ewelina Węglarz-Tomczak
University of Amsterdam
Amsterdam, the Netherlands
e.m.weglarz-tomczak@uva.nl

Agoston E. Eiben
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
a.e.eiben@vu.nl

ABSTRACT

Differential evolution (DE) is a well-known type of evolutionary algorithms (EA). Similarly to other EA variants it can suffer from small populations and loose diversity too quickly. This paper presents a new approach to mitigate this issue: We propose to generate new candidate solutions by utilizing reversible linear transformations applied to a triplet of solutions from the population. In other words, the population is enlarged by using newly generated individuals without evaluating their fitness. We assess our methods on three problems: (i) benchmark function optimization, (ii) discovering parameter values of the gene repressilator system, (iii) learning neural networks. The empirical results indicate that the proposed approach outperforms vanilla DE and a version of DE with applying differential mutation three times on all testbeds.

CCS CONCEPTS

• **Theory of computation → Bio-inspired optimization**;

KEYWORDS

Black-box optimization, reversible computation, population-based algorithms

ACM Reference Format:
Jakub M. Tomczak, Ewelina Węglarz-Tomczak, and Agoston E. Eiben. 2020. Differential Evolution with Reversible Linear Transformations. In *Proceedings of the Genetic and Evolutionary Computation Conference 2020 (GECCO'20 Companion).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389972

1 INTRODUCTION

Black-box optimization. We consider an optimization problem of a function $f : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X} \subseteq \mathbb{R}^D$ is the search space. In this paper we focus on the minimization problem, namely:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x}). \tag{1}$$

Further, we assume that the analytical form of the function f is unknown or cannot be used to calculate derivatives, however, we can query it through a simulation or experimental measurements. Problems of this sort are known as *black-box optimization problems* [1, 5]. Additionally, we consider a bounded search space, *i.e.*, we include inequality constraints for all dimensions in the form: $l_d \leq x_d \leq u_d$, where $l_d, u_d \in \mathbb{R}$ and $l_d < u_d$, for $d = 1, 2, \ldots, D$.

GECCO'20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07…$15.00
https://doi.org/10.1145/3377929.3389972

Differential Evolution. A well-known method for black-box optimization problems is *differential evolution* (DE) [8] that requires a population of candidate solutions, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, to iteratively generate new query points. A new candidate is generated by randomly picking a triple from the population, $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{X}$, and $\mathbf{x}_i$ is perturbed by adding a scaled difference between $\mathbf{x}_j$ and $\mathbf{x}_k$:

$$\mathbf{y} = \mathbf{x}_i + F(\mathbf{x}_j - \mathbf{x}_k), \tag{2}$$

where $F \in \mathbb{R}_+$ is the scaling factor. This operation could be seen as an adaptive *mutation operator*, *i.e*, *differential mutation* [7].

Further, the authors of [8] proposed to sample a binary mask $\mathbf{m} \in \{0, 1\}^D$ according to the Bernoulli distribution with probability $p = P(m_d = 1)$ shared across all D dimensions, and calculate the final candidate according to the following formula:

$$\mathbf{v} = \mathbf{m} \odot \mathbf{y} + (1 - \mathbf{m}) \odot \mathbf{x}_i, \tag{3}$$

where $\odot$ denotes the element-wise multiplication. In the evolutionary computation literature this operation is known as *uniform crossover operator* [2, 4]. In this paper, we fix $p = 0.9$ following [6] and use the uniform crossover in all methods.

The last component of a population-based method is a selection mechanism. Here we combine the old population with the new one and select N candidates with highest fitness values (*i.e.*, the deterministic $(\mu + \lambda)$ selection) [2, 4].

This variant of DE is referred to as "DE/*rand*/*1*/*bin*", where *rand* stands for randomly selecting a base vector, *1* is for adding a single perturbation (a vector difference) and *bin* denotes the uniform crossover. Sometimes it is called *classic DE* [7].

2 OUR APPROACH

Generating new candidates in DE requires sampling a triplet of solutions and, based on these points, one solution is perturbed using the other two solutions. This approach possesses multiple advantages, naming only a few:

(i) it is non-parametric, *i.e.*, contrary to evolutionary strategies [3], no assumption on the underlying distribution of the population is made;
(ii) it has been shown to be effective in many benchmark optimization problems and real-life applications [7].

However, the number of possible perturbations is finite and relies entirely on the population size. Therefore, a small population size could produce insufficient variability of new candidate solutions. To counteract this issue, we propose the following solutions:

(1) In order to increase variability, we can perturb candidates multiple times by running the differential mutation more than once (*e.g.*, three times).
(2) In fact, we can use the selected triple of points and use it three times to generate new points. In other words, we notice that there is no need to sample three different triplets.

(3) We propose to modify the selected triplet by using generated new solutions *on-the-fly*. This approach allows to enlarge the population size.

Differential Evolution x3. In the first approach we generate a larger new population by perturbing the point $\mathbf{x}_i$ using multiple candidate solutions, namely, $\mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l, \mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_q \in \mathcal{X}$. Then, we can produce $3N$ new candidate solutions instead of N as follows:

$$\mathbf{y}_1 = \mathbf{x}_i + F(\mathbf{x}_j - \mathbf{x}_k) \tag{4}$$
$$\mathbf{y}_2 = \mathbf{x}_i + F(\mathbf{x}_l - \mathbf{x}_m) \tag{5}$$
$$\mathbf{y}_3 = \mathbf{x}_i + F(\mathbf{x}_n - \mathbf{x}_q). \tag{6}$$

This approach requires sampling more pairs and evaluating more points, however, it allows to better explore the search space. We refer to this approach as *Differential Evolution* ×3, or DEx3 for short.

Antisymmetric Differential Evolution. We first notice that in the DEx3 approach we sample three pairs of points to calculate perturbations. Since we pick them at random, we propose to sample three candidates $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathcal{X}$ and calculate perturbations by changing their positions only, that is:

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{x}_i + F(\mathbf{x}_j - \mathbf{x}_k)\\ \mathbf{y}_2 &= \mathbf{x}_j + F(\mathbf{x}_k - \mathbf{x}_i)\\ \mathbf{y}_3 &= \mathbf{x}_k + F(\mathbf{x}_i - \mathbf{x}_j).\end{aligned} \tag{7}$$

In other words, we perturb each point by using the remaining two. Interestingly, we notice that Eq. 7 corresponds to applying a linear transformation to these three points. For this purpose, we rewrite (7) using matrix notation by introducing matrices $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3]^\top$ and $\mathbf{X} = [\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k]^\top$ that yields $\mathbf{Y} = \mathbf{M}\mathbf{X}$, where:

$$\mathbf{M} = \underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}_{\mathbf{I}} + \underbrace{\begin{bmatrix}0 & F & -F\\ -F & 0 & F\\ F & -F & 0\end{bmatrix}}_{\mathbf{A}}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{A}$ is the antisymmetric matrix.

Comparing Eq. 7 to DE×3 we notice that there is no need to sample additional candidates beyond one triplet. Moreover, the new mutation in (7) allows us to analyze the transformation from the algebraic perspective. We refer to this version of DE as *Antisymmetric Differential Evolution* (ADE), because the linear transformation consists of the identity matrix and the antisymmetric matrix parameterized with the scaling factor F.

Reversible Differential Evolution. The linear transformation presented in Eq. 7 allows to utilize the triplet $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ to generate three new points, however, it could be still seen as applying DE three times, but in a specific manner (*i.e.*, by defining the linear operator $\mathbf{M}$). A natural question arises whether a different transformation could be proposed that allows *better* exploitation and/or exploration of the search space. The mutation operator in DE perturbs candidates using other individuals in the population. As a result, having too small population could limit exploration of the search space. In order to overcome this issue, we propose to modify Using new candidates $\mathbf{y}_1$ and $\mathbf{y}_2$ allows to calculate perturbations using points outside the population. This approach does not follow ADE by using newly generated candidates *on-the-fly*, that is:

$$\begin{aligned}\mathbf{y}_1 &= x_i + F(\mathbf{x}_j - \mathbf{x}_k)\\ \mathbf{y}_2 &= x_j + F(\mathbf{x}_k - \mathbf{y}_1)\\ \mathbf{y}_3 &= x_k + F(\mathbf{y}_1 - \mathbf{y}_2).\end{aligned} \tag{9}$$

a typical construction of an EA where only evaluated candidates are mutated. Further, similarly to ADE, we can express (9) as a linear transformation $\mathbf{Y} = \mathbf{R}\mathbf{X}$ with the following linear operator:

$$\mathbf{R} = \begin{bmatrix}1 & F & -F\\ -F & 1-F^2 & F+F^2\\ F+F^2 & -F+F^2+F^3 & 1-2F^2-F^3\end{bmatrix}. \tag{10}$$

In order to obtain the matrix $\mathbf{R}$, we need to plug $\mathbf{y}_1$ to the second and third equation in (9), and then $\mathbf{y}_2$ to the last equation in (9). We refer to this version of DE as *Reversible Differential Evolution* (RevDE), because the linear transformation is reversible.

3 EXPERIMENTS

In order to verify our approach empirically, we compare the three proposed methods and the standard DE on three testbeds:

(1) *Benchmark functions*: selected benchmark function for optimization.
(2) *Gene Repressilator System*: discovering parameter values of a system of ordinary differential equations for given observations.
(3) *Neural Networks Learning*: learning a neural network with one hidden layer on image dataset.

Results. In all benchmark test cases, RevDE achieved the best results in terms of both final objective value and convergence speed. This result is remarkable, because new candidate solutions are generated *on-the-fly* and are used to generate to new points. In the gene repressilator model RevDE converged faster than ADE and DEx3. In learning neural networks ADE performed slightly better than RevDE, but overall ADE and RevDE outperformed DEx3 significantly. The results and more thorough discussions of all experiments are available in the long version of this paper: https://arxiv.org/abs/2002.02869.

The code of the methods and all experiments is available under the following link: https://github.com/jmtomczak/reversible-de.

REFERENCES

[1] Charles Audet and Warren Hare. 2017. *Derivative-free and Blackbox Optimization.* Springer.
[2] Thomas Bäck. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford university press.
[3] Thomas Bäck, Christophe Foussette, and Peter Krause. 2013. *Contemporary Evolution Strategies.* Springer.
[4] Agoston E. Eiben and James E. Smith. 2015. *Introduction to Evolutionary Computing.* Vol. 53. Springer.
[5] Donald R. Jones, Matthias Schonlau, and William J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.
[6] Magnus Erik Hvass Pedersen. 2010. *Good parameters for differential evolution.* Technical Report HL1002. Hvass Laboratories.
[7] Kenneth Price, Rainer M Storn, and Jouni A. Lampinen. 2006. *Differential Evolution: A Practical Approach to Global Optimization.* Springer Science & Business Media.
[8] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 4 (1997), 341–359.